# Интеграция АБИС «ИРБИС64» с электронными картами DiCARDS

Приложения для интеграции АБИС «ИРБИС64» с DiCARDS предоставляют следующие возможности:

* Регистрация читателей библиотеки, имеющих регистрацию в «ИРБИС64», в системе карт DiCARDS (выполняется сотрудником библиотеки по запросу читателя).
* Автоматизированная регистрация клиентов системы DiCARDS в базе читателей «ИРБИС64».
* Автоматическая рассылка push-уведомлений на смартфоны читателей, зарегистрированных в DiCARDS и имеющим задолженность перед библиотекой.
* Автоматическая актуализация списка книг, числящихся выданными читателю, зарегистрированному в DiCARDS, на электронной карте в смартфоне читателя.

Приложения для интеграции созданы на основе технологии Microsoft .NET Framework и предъявляют следующие требования:

* Операционная система Microsoft Windows версий 7 (с установленным SP1), 8, 8.1, 10, Server 2008 R2, Server 2012, Server 2012 R2, Server 2016 и Server 2019. Редакции Home Server и Nano Server не поддерживаются.
* Установленный Microsoft .NET Framework 4.6.1 или более новая версия. Редакция .NET Core не поддерживается.
* Не менее 400 Мб свободного места на диске, на который будет выполняться установка.

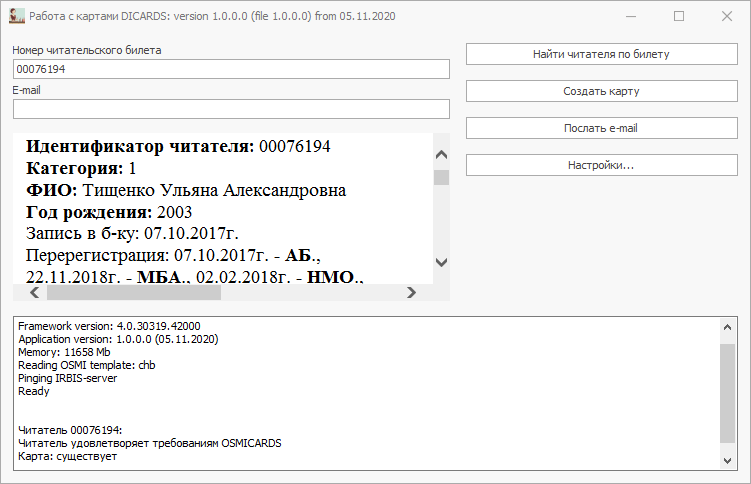
Предусмотрено (настраиваемое) логирование действий программы в текстовый файл в директории **%ProgramData%\Dicards**. Лог-файлы могут потребовать дополнительного свободного места на системном диске.

# Программа «Front Office»

Программа предназначена для регистрации читателя, сведения о котором ранее были внесены в базу данных системы «ИРБИС64», в системе электронных карт DiCARDS. Программа устанавливается на рабочем месте библиотекаря в отделе регистрации читателей. Сотрудник, работающий с программой, должен быть аккредитован для работы с персональными данными.

Алгоритм работы с данной программой таков:

1. Читатель, уже имеющий билет данной библиотеки (неважно, традиционный или электронный), обращается в отдел регистрации и предъявляет свой билет.
2. Библиотекарь считывает штрих-код билета или вносит его с клавиатуры в поле «Номер читательского билета».
3. Для регистрации в системе DiCARDS читатель должен иметь адрес электронной почты. Программа проверит его наличие, и, если его нет, откажется регистрировать читателя. Если адрес просто не внесен в базу данных «ИРБИС64», то достаточно ввести его в поле «E-mail».
4. Два читателя не могут регистрироваться на один e-mail, т. к. система рассчитана на уникальные адреса электронной почты.
5. Если запись в базе данных «ИРБИС» удовлетворяет условиям системы, можно создать карту в системе DiCARDS при помощи кнопки «Создать карту».
6. Если карта успешно создана, необходимо отправить письмо на адрес читателя с помощью кнопки «Послать e-mail». Письма можно посылать столько раз, сколько понадобится. Рассылка писем производится серверами системы DiCARDS.



Настройка системы (см. снимок экрана ниже) доступна в любой момент при помощи кнопки «Настройка…» и требует знания пароля. Пароль – слово **cards** в нижнем регистре строчными латинскими буквами. При необходимости его можно поменять в файле **FrontOffice.exe.config**. Лучше, если пароль будет знать только администратор системы.

Настройки программы для связи с серверами DiCARDS и ИРБИС64 хранятся в файле **dicards.json** рядом с исполняемым файлом. Формат настроек JSON.

{

"connectionString": "host=127.0.0.1;port=6666;user=libra;password=secret;db=RDR;",

"apiID": "PPXT0TUWRVIB3M15048X",

"apiKey": "008415c3728b22dce016dd9a5af9f09fadbe23b0",

"baseUri": "https://api.osmicards.com/v2",

"template": "chb",

"readerID": "30",

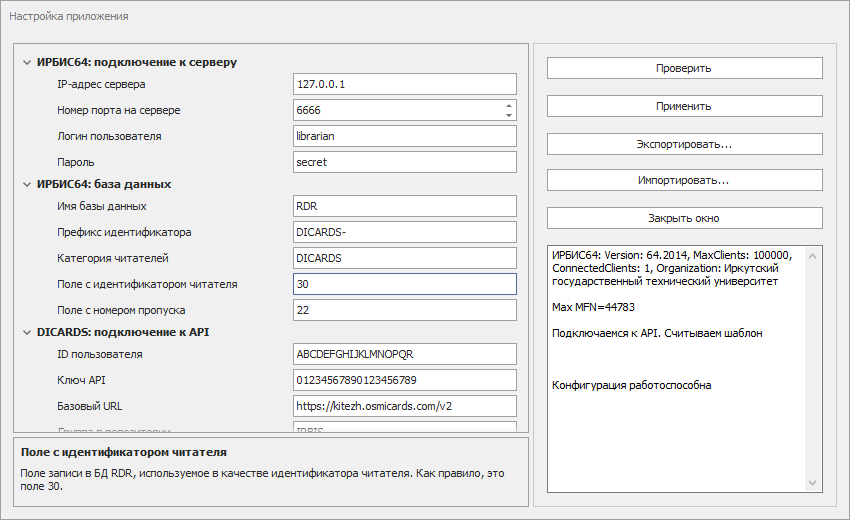
"ticket": "22"

}

Настройки по пунктам:

* **connectionString** – строка подключения к серверу ИРБИС64. В ней обязательно должны быть перечислены: IP-адрес хоста, на котором запущен сервер ИРБИС64, номер порта, логин пользователя, под которым программа будет входить на сервер, пароль и имя базы данных, в которой хранятся данные о читателях.
* **apiID** – идентификатор пользователя системы DiCARDS.
* **apiKey** – пароль для доступа к API DiCARDS.
* **baseUri** – URL точки подключения к API DiCARDS.
* **template –** имя шаблона карточки, которая будет послана на смартфон читателя. Эту карту должен создать администратор, занимающийся подключением библиотеки к системе DiCARDS.
* **readerID –** метка поля в базе данных читателей, в котором хранится идентификатор читателя. По умолчанию это поле 30.
* **ticket ­–** меткаполя в базе данных читателей, в котором хранится номер пропуска (читательского билета). Часто в этом качестве выступает RFID-метка пропуска в библиотеку. По умолчанию это поле 22.

Формат настроек **dicards.json** идентичен для программ Front Office и Back Office (о ней ниже), так что их можно переносить между программами, например, с помощью кнопок «Экспортировать…» и «Импортировать…» в диалоге «Настройка приложения». При экспортировании настроек чувствительные данные (например, ключ доступа к API) шифруются. Это шифрование прозрачно для пользователя и не требует от него никаких действий.



Настройки логирования действий хранятся в файле **NLog.config** рядом с исполняемым файлом. Формат настроек XML (см. пример ниже).

<?xml version="1.0" encoding="utf-8" ?>

<nlog xmlns="http://www.nlog-project.org/schemas/NLog.xsd"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<variable name="relaxed"

value="${longdate} | ${level:uppercase=true:padding=-5} | ${message}"

/>

<targets>

<target name="logconsole" xsi:type="ColoredConsole"

layout="${relaxed}"

/>

<target name="debuglog" xsi:type="File"

layout="${relaxed}"

fileName="LogFile.txt"

/>

<target name="productionlog" xsi:type="File"

layout="${relaxed}"

fileName="${specialfolder:folder=CommonApplicationData}/Dicards/Front Office/LogFile.txt"

/>

</targets>

<rules>

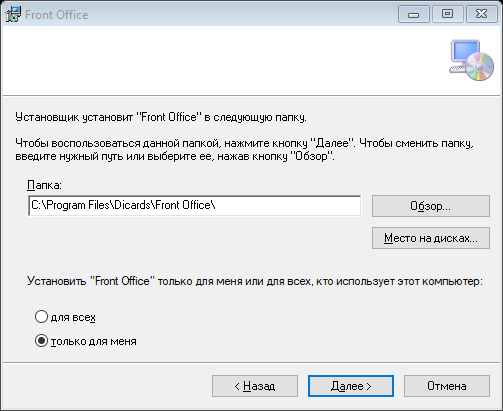
<logger name="\*" minlevel="Info" writeTo="productionlog" />

</rules>

</nlog>

Согласно этим настройкам, логи пишутся в файл **%ProgramData%\Dicards\Front Office\LogFile.txt**. При необходимости этот файл можно безболезненно удалять, программа создаст его снова. Подробнее о настройке NLog можно прочитать здесь: <https://nlog-project.org/config/>

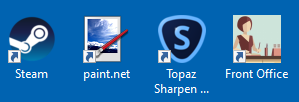
Установку программы **Front Office** рекомендуется осуществлять с помощью инсталлятора **FrontSetup.msi**.



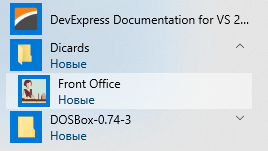
Для установки в папку внутри **%ProgramFiles%** требуются права администратора, при их отсутствии можно установить программу в **%LOCALAPPDATA%\Dicards\Front Office** (проще говоря, в **C:\Users\username\AppData\Local\Dicards\Front Office**). Такая установка не требует никаких дополнительных прав и никак не сказывается на работе программы.

После распаковки файлов в указанную папку инсталлятор показывает окно для ввода конфигурации. Очень важно заполнить все поля в этом окне и убедиться в работоспособности, нажав кнопку «Проверить», а затем «Применить» (после успешной проверки). Не сконфигурированная программа не будет работать.

Инсталлятор создает ярлык на рабочем столе



и папку в меню «Пуск»



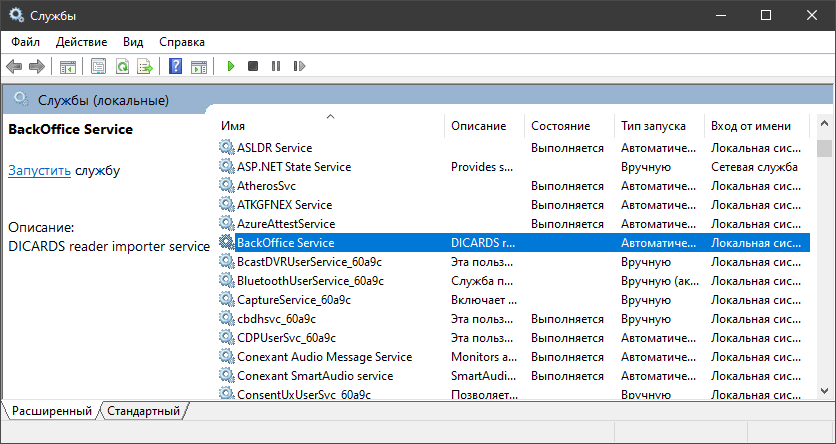
Использовать можно любой из этих способов запуска программы, они идентичны.

# Программа «Back Office»

Программа «Back Office» предназначена для выполнения периодических фоновых задач, таких как перенос сведений о зарегистрированных пользователях из системы DiCARDS в базу данных ИРБИС64.

Программа реализована в виде службы Windows и должна устанавливаться на серверную операционную систему, которая будет работать круглосуточно.

Имя службы **BackOffice Service**.



Предусмотрено три фоновых задачи:

* Перенос сведений о пользователях, зарегистрировавшихся в системе DiCARDS, в базу данных ИРБИС64. Эта задача автоматически выполняется раз в пять минут, так что читателю не потребуется долго ждать регистрации.
* Рассылка push-уведомлений о просроченных книгах на смартфон читателя. Эта задача автоматически выполняется раз в сутки.
* Автоматическая актуализация списка книг, числящихся за читателем на карте в приложении Wallet в смартфоне читателя. Эта задача также отрабатывает раз в сутки.

При необходимости периодичность любой задачи можно поменять. Кроме того, любая из этих задач может быть отключена. Настройки задач хранятся в файле **jobs.json** рядом с исполняемым файлом программы. Формат настроек JSON.

{

"jobs": [

{

"name": "Importer",

"group": "Dicards",

"description": "Reader importer job",

"enabled": true,

"type": "BackOffice.Jobs.ImporterJob, BackOffice",

"cron": "\* 0/5 \* \* \* ?"

},

{

"name": "Pusher",

"group": "Dicards",

"description": "Message pushing job",

"enabled": false,

"type": "BackOffice.Jobs.PushJob, BackOffice",

"cron": "0 30 14 \* \* ?"

},

{

"name": "Reminder",

"group": "Dicards",

"description": "Reminder job",

"enabled": false,

"type": "BackOffice.Jobs.RemindJob, BackOffice",

"cron": "0 0 14 \* \* ?"

}

]

}

Настройки каждой задачи по пунктам:

* **name** – уникальный идентификатор задания.
* **group** – идентификатор группы заданий. Рекомендуется использовать значение **Dicards**.
* **description** – произвольное описание задания.
* **enabled** – служит для отключения ненужных в данный момент заданий. При настройках по умолчанию задания **Pusher** и **Reminder** отключены.
* **type** – полное имя класса задания и ссылка на сборку, в которой находится этот класс.
* **cron** – выражение в формате cron, задающее периодичность исполнения задания.

По умолчанию зарегистрированы три задачи:

* Importer – импорт пользователей из репозитория DiCARDS.
* Pusher – рассылка push-уведомлений.
* Reminder – актуализация списков «литература у меня на руках».

Подробнее о cron-выражениях можно прочитать здесь: <https://ru.wikipedia.org/wiki/Cron>

В **Back Office** применяется следующий формат cron-выражения:

\* \* \* \* \* \*

- - - - - -

| | | | | |

| | | | | ----- день недели (SUN, MON и т. д.)

| | | | ------- месяц (1—12)

| | | --------- день (1—31)

| | ----------- час (0—23)

| ------------- минута (0—59)

|-------------- секунда (0-59)

После внесения любых изменений в файл **jobs.json** необходимо перезапустить службу.

Настройки для связи с сервером ИРБИС64 и API DiCARDS полностью аналогичны описанным выше настройкам в программе **Front Office**. Однако используются несколько дополнительных настроек (выделены жирным)

{

"connectionString": "host=127.0.0.1;port=6666;user=libra;password=secret;db=RDR;",

"apiID": "PPXT0TUWRVIB3M15048X",

"apiKey": "008415c3728b22dce016dd9a5af9f09fadbe23b0",

"baseUri": "https://api.osmicards.com/v2",

**"group": "IRBIS",**

**"prefix": "DICARDS-",**

**"category": "DICARDS",**

"template": "chb",

**"message": "Пора сдавать книги в библиотеку",**

}

* **group** – группа регистрации новых пользователей (задает DiCARDS).
* **prefix** – префикс, присваиваемый номеру читательского билета для импортированного читателя, может быть пустой строкой.
* **category** – категория, присваиваемая импортируемому читателю (задается самой библиотекой, исходя из её задач).
* **fioField** – имя поля в шаблоне карточки читателя, в которое будет помещено ФИО читателя. Имя поля задает сама библиотека, например «ФИО», оно будет видно в приложении.
* **reminderField** – имя поля в шаблоне карточки читателя, в которое будет помещено напоминание о задолженности перед библиотекой. Имя поля задает сама библиотека, например «НАПОМИНАНИЕ», оно будет видно в приложении.
* **reminderMessage** – сообщение отправляемое задачей Pusher читателям, имеющим просроченную задолженность (одно сообщение на всех читателей).
* **totalCountField** – имя поля в шаблоне карточки читателя, в которое будет помещено общее количество документов, числящихся за читателем. Имя поля задает сама библиотека, например «ВСЕГО», оно будет видно в приложении.
* **totalCountFormat** – имя файла формата, используемого для формирования значения в поле **totalCountField**. По умолчанию это “|total\_count.pft”. Здесь символ “|” означает, что файл находится в локальной файловой системе (рядом с программой **Back Office**), а не на сервере ИРБИС64.
* **expiredCountField** – имя поля в шаблоне карточки читателя, в которое будет помещено количество документов, просроченных читателем. Имя поля задает сама библиотека, например «ДОЛГ», оно будет видно в приложении.
* **expiredCountFormat** – имя файла формата, используемого для формирования значения в поле **expiredCountField**. По умолчанию это “|expired\_count.pft”. Здесь символ “|” означает, что файл находится в локальной файловой системе.
* **totalListField** – имя поля в шаблоне карточки читателя, в которое будет помещен общий список документов, числящихся за читателем. Имя поля задает сама библиотека, например «КНИГИ», оно будет видно в приложении.
* **totalListFormat** – имя файла формата, используемого для формирования значения в поле **totalListField**. По умолчанию это “|total\_list.pft”. Здесь символ “|” означает, что файл находится в локальной файловой системе.
* **expiredListField** – имя поля в шаблоне карточки читателя, в которое будет помещен список документов, просроченных читателем. Имя поля задает сама библиотека, например «ПРОСРОЧЕННЫЕ КНИГИ», оно будет видно в приложении.
* **expiredListFormat** – имя файла формата, используемого для формирования значения в поле **expiredListField**. По умолчанию это “|expired\_list.pft”. Здесь символ “|” означает, что файл находится в локальной файловой системе.

Напоминаем, что формат файла **dicards.json** у программ **Back Office** и **Front Office** полностью совпадает.

Настройки логирования хранятся в файле **NLog.config**, они полностью аналогичны ранее описанным настройкам программы **Front Office**. Можно настроить выгрузку логов в системный журнал Windows Event Log, для этого достаточно зарегистрировать соответствующий источник событий в командной строке (с правами администратора) следующей командой (естественно, введенной в одну строку):

eventcreate /ID 1 /L APPLICATION /T INFORMATION /SO BackOffice /D "Created log for BackOffice"

и раскомментировать в **NLog.config** логгер «eventlog». После перезапуска службы события будут отражаться в системном приложении «Просмотр событий».

Изображение выглядит как текст

Автоматически созданное описание

Установка службы **Back Office** производится инсталлятором **BackSetup.msi** только от имени администратора! После распаковки файлов инсталлятор запросит конфигурацию для службы. Её необходимо заполнить полностью и правильно, иначе последующая регистрация службы может оказаться неудачной, и установка будет отменена.

После успешной установки службы рекомендуется запустить системное приложение «Службы» и убедиться, что служба **Back Office** появилась в нем и успешно запускается.

Ошибки, возникающие при установке, записываются в текстовый файл **%ProgramData%\Dicards\Back Office\LogFile.txt**. Они выглядят примерно так:

Невозможно запустить службу. AM.VerificationException: Выдано исключение типа "AM.VerificationException".

в RestfulIrbis.OsmiCards.DicardsConfiguration.Verify(Boolean throwOnError) в D:\Projects\ManagedClient.45\Source\Classic\Libs\RestfulIrbis\Source\OsmiCards\DicardsConfiguration.cs:строка 215

в BackOffice.Importer.LoadConfiguration() в D:\Projects\ManagedClient.45\Source\Classic\Dicards\BackOffice\Source\Importer.cs:строка 137

в BackOffice.DicardsService.Start(HostControl hostControl) в D:\Projects\ManagedClient.45\Source\Classic\Dicards\BackOffice\Source\DicardsService.cs:строка 52

в Topshelf.Builders.ControlServiceBuilder`1.ControlServiceHandle.Start(HostControl hostControl)

в Topshelf.Runtime.Windows.WindowsServiceHost.OnStart(String[] args)

в System.ServiceProcess.ServiceBase.ServiceQueuedMainCallback(Object state)

При возникновении ошибок необходимо собрать следующую информацию:

* Вышеописанные логи.
* Файл **dicards.json**.
* Файл **jobs.json**.

и отправить их разработчику.

Параметры командной строки **BackOffice.exe**:

BackOffice <команда>

где **<команда>** одна из следующих:

* **configure** – диалог настройки связи с сервером ИРБИС64 и API DiCARDS.
* **run** – однократное выполнение всех разрешенных задач из **jobs.json**.
* **install** – регистрация службы в системе (нужны права администратора).
* **uninstall** – отмена регистрации службы (нужны права администратора).
* **start** – запуск службы (нужны права администратора).
* **stop** – останов службы (нужны права администратора).

Будучи запущенной без указания команды, программа **BackOffice.exe** работает как простое консольное приложение.